Evaluation of MBARI PUCK protocol for interoperable ocean observatories

Joaquin del Rio[1], Tom O'Reilly[2], Kent Headley[2], Daniel Mihai Toma[1], Neil Cater[4], Carlos Rueda[2],, Duane Edgington[2], Chris Ng[5] , Ikram Bghiel[1], Jesper Zedlitz[6], Frank Johnson[3], Greg Johnson[3], Eric Davis[4], Reo Phillips[5], Sameer Tilak[9], Eric Delory[7], Antoni Manuel[1], Christoph Waldman[8]

[1]SARTI, Technical University of Catalonia, Joaquin.del.rio@upc.edu, [2]MBARI, oreilly@mbari.org, [3]RBR, [4]Fisheries and Marine Institute of Memorial University of Newfoundland, [5]AXYS, [6]Kiel University, [7]dBScale, [8]MARUM University, [9]University of California at San Diego

Abstract- *IEEE-1451[1] and OGC Sensor Web Enablement (OGC SWE)[2] define standard protocols to operate instruments, including methods to calibrate, configure, trigger data acquisition, and retrieve instrument data based on specified temporal and geospatial criteria. These standards also provide standard ways to describe instrument capabilities, properties, and data structures produced by the instrument. These standard operational protocols and descriptions enable observing systems to manage very diverse instruments as well as to acquire, process, and interpret their data in a uniform and automated manner. We refer to this property as "instrument interoperability". This paper describes integration and evaluation of MBARI PUCK protocol [3] within different observatories including OBSEA [4,5] in Spain, the ESONET test-bed in Germany, and the SmartBay observatory in Canada.*

Keywords- MBARI PUCK Protocol, Instrument Interoperability, IEEE1451, OGC SWE

INTRODUCTION

To achieve instrument interoperability, the physical instrument must be reliably associated with software and information that conform to standard protocols and descriptions. In most cases today, the "firmware" that is physically embedded within the instrument does not conform to standards; instead standards-compliant external instrument "driver" software and metadata files residing on observatory host computers are logically associated with the physical instruments. Setting up the logical association is typically a manual process; technicians must install instrument driver software on the host, specify a host data port where the instrument is installed, and specify baud rates, configuration files, and so on. This manual configuration process can be tedious, time-consuming, and hence prone to human error. Moreover the configuration process must sometimes be performed aboard ships and buoys under severe environmental conditions that challenge human physiology and psychology, thus increasing the chances for error.

An alternative approach is to embed the standards physically within the instrument; in this case the instrument will respond appropriately to standard operational protocols, and will supply descriptive information in standard format. Thus the observing system can automatically identify the instrument and utilize the instrument and its data when it is physically installed, and there is no need for technicians to manually set up a logical association between physical instrument and host drivers and configuration files. There are several challenges to this approach that can be solved by using standards such as IEEE1451, OGC SWE and MBARI PUCK protocol described below.

IEEE-1451 and OGC SWE

IEEE-1451 and OGC SWE are rather complex, which is to be expected as these standards are also quite comprehensive. This complexity presents challenges for instrument manufacturers who must thoroughly understand the standard and who must correctly implement it in firmware. Moreover embedded instrument processors are often designed for low cost and low-power environments, and hence may not be capable of fully implementing the standards. Another drawback is that manufacturers would likely have to abandon existing instrument firmware that does not implement the standard; this existing firmware often represents a very considerable investment by the manufacturer. A third drawback is that IEEE-1451 and OGC SWE are still evolving, again due to the comprehensive nature of these standards. Thus either the standard revision process must be very carefully managed to ensure "backwards compatibility", or instrument firmware must be occasionally upgraded to remain compliant with the latest standard. Both of these alternatives present non-trivial challenges to instrument manufacturers and standards bodies.

MBARI PUCK Protocol

A third approach is implemented by MBARI PUCK protocol, which does not itself implement interoperability, but rather provides the lower tier in a hierarchy of standards that achieve this goal. PUCK defines a simple standard embedded instrument protocol to store and retrieve information from the instrument. The information consists of a minimal instrument datasheet that includes a universally unique instrument serial number, a manufacturer ID, and a small amount of other metadata PUCK protocol also allows an optional "payload" consisting of any information needed by a particular observing system. The payload format and content are not constrained by PUCK protocol, and can include executable driver code that implements a standard operating protocol as well as metadata that describe the instrument in a standard way. Using PUCK protocol, technicians can store payload contents with the instrument before deployment. When the instrument is deployed, payload is retrieved by the host and utilized appropriately; e.g. the host can execute the driver code, and can use or distribute the standard metadata to other locations on the network. Thus standard IEEE-1451 and OGC SWE components can be automatically retrieved and installed by the host when a PUCK-enabled instrument is plugged in, overcoming the difficulties of manual installation. PUCK protocol is simple, and readily implemented in even simple instrument processors; several manufacturers now implement MBARI PUCK protocol in their instruments, and report just a few weeks of engineering effort to do so. PUCK protocol augments rather than replaces existing instrument protocols, and manufactures can usually implement PUCK by extending their existing protocol rather than starting from scratch. Since the protocol is simple, it is likely to be stable, so manufacturers to do not have to modify firmware to keep up with an evolving standard. As higher-level IEEE-1451 and OGC SWE standards evolve, the instrument PUCK payloads can simply be updated through PUCK protocol. The PUCK protocol specification is available at http://www.mbari.org/pw.

PUCK INTEGRATION

Until recently, PUCK protocol was used exclusively on MBARI moored and cable-to-shore observatories [6]. We describe tests to integrate and evaluate the protocol on several non-MBARI systems, including ESONET test-bed observatories such as OBSEA [3,4] and the SmartBay observatory in Canada. We estimate the engineering effort required to integrate PUCK into these systems, and summarize the benefits gained for that effort. We discuss possible refinements to the protocol and describe plans to submit MBARI PUCK as a formal standard.

PUCK INTEGRATION AT WESTERN MEDITERRANEAN OBSERVATORY, OBSEA, SPAIN

At OBSEA Observatory, two CTD are been used to test the integration of PUCK protocol. Theses instruments were a RBR CTD with PUCK implemented in firmware and a Seabird CTD with an external PUCK hardware. Integration starts by developing the instrument metadata. Two different metadata files were implemented for each instrument: a SensorML file and a XML IEEE 1451 TEDS file. These files are stored in the PUCK payload memory. Each file is preceded by a tag that specifies the file type, as shown in Table 1 (the tag format and attributes will be proposed as an addendum to the PUCK version 1.3 specification)

Payload Type	Description
IEEE-1451- binary-TEDS	IEEE-1451 TEDS (binary format)
IEEE-1451-xml-TEDS	IEEE-1451 TEDS (XML format)
SWE-SensorML	SensorML format
MBARI-SIAM	MBARI SIAM JAR file

Table 1. Recommended Payload type name

A web-based tool is being developed to simplify creation of SensorML and IEEE 1451 TEDS files for specific instruments, using consistent syntax and attribute names. The user indicates the structure of the sensor system (system type, variables, and subsystems) while being able to choose URIs (*Uniform Resource Identifiers*) via drop-down lists containing standard entries for sensor types and variables, and then the tool generates the resulting document. The drop-down lists are populated with definitions registered in the MMI Ontology Registry and Repository, ORR, http://mmisw.org/orr [7]. Figure 1 illustrates the basic interaction with the definition of an output variable. The user clicks a button to select an appropriate definition from the NetCDF Climate and Forecast (CF) Metadata Convention standard name vocabulary (http://cf-pcmdi.llnl.gov/). A similar selection mechanism is available for sensor types. The tool allows the description to include nested subsytems, each with the corresponding variables. Once the desired structure has been completed, the "Generate SensorML/TEDS" button creates a file that can then be stored in an instrument's PUCK payload.

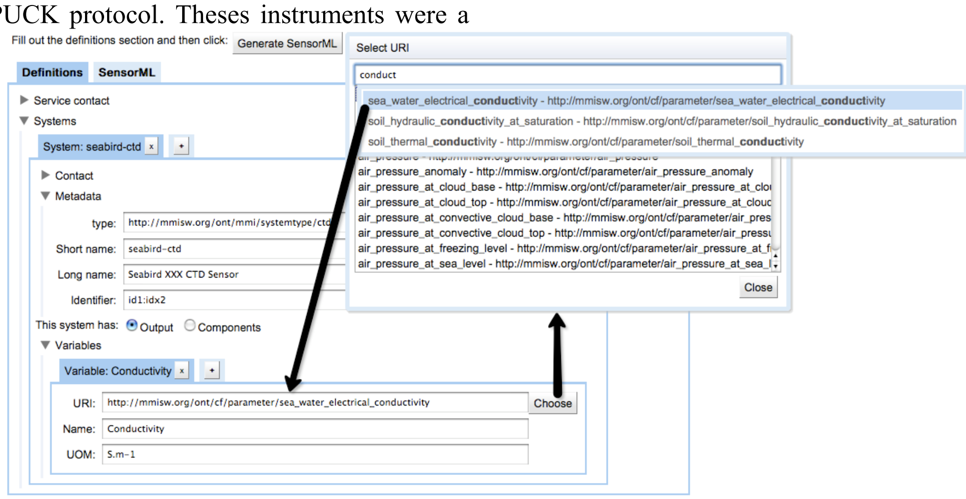

Figure 1. Web interface to generate SensorML PUCK Payload

The communication between instruments (in this case 2 CTDs) and the NCAP host computer is implemented by a serial RS232 link. The host computer is running an IEEE1451.0 HTTP server and an automatic instrument recognition algorithm to automatically detect a new instrument plugged into a serial port. This detection protocol is shown in figure 2. The host computer periodically interrogates the serial port for a PUCK-enabled instrument. When the host receives a PUCK response from the serial port, the host retrieves the 96-byte PUCK datasheet and examines the UUID to determine if a new instrument has been installed. If so, the host retrieves the SensorML and IEEE 1451 TEDS description from the instrument's PUCK payload, and loads an appropriate driver. Finally the driver retrieves a new data sample from the instrument. These operations are performed at the sampling frequency specified for the instrument.

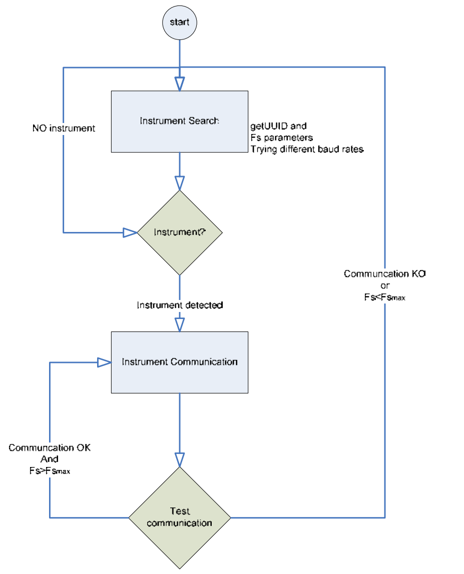

Figure 2. Automatic Instrument Recognition protocol

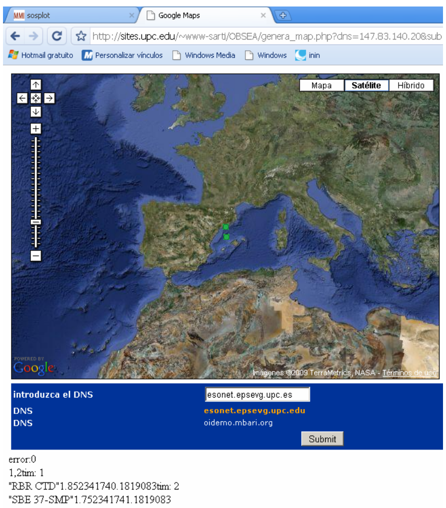

Figure 3. Google Maps application to show instrument availability

The IEEE1451.0 HTTP server running on the NCAP host computer keeps track of instruments or TIMs connected to the NCAP serial ports. A web application based in Google Maps retrieves the information from the NCAP using IEEE1451.0 commands such as "http://esonet.epsevg.upc.es:1451/1451/Discovery/TIMDiscovery?ncapId=4&responseFormat=xml"and ReadTIMGeoLocationTEDS command in order to mark the position of the instrument in the Map as is shown in Figure 2.

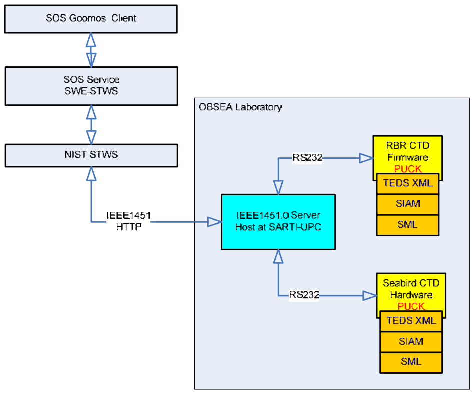

Figure 4. Block Diagram of the Test Bench

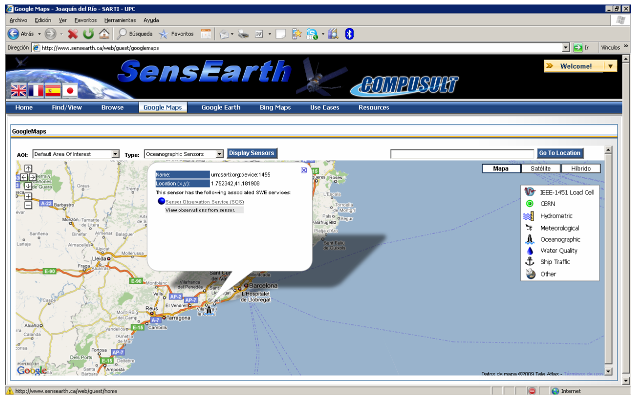

Figure 5. SOS Client from Compusult

In addition a Sensor Observation Service (SOS) runs on the NCAP host computer, in parallel with the IEEE1451.0 server. This SOS updates its properties about the number of instruments connected to the host. An SOS client such as Compusult's SenseEarth (http://sensearth.ca/) retrieves the SensorML instrument description originally stored in the instrument PUCK, thereby visualizing information geographically in a Google Maps application and reading data from the instruments. Figure 4 shows the schema of the instruments and services running the SOS and Figure 5 shows a Compusult SOS client used to visualize real-time data.

PUCK INTEGRATION AT SMARTBAY OCEAN OBSERVING SYSTEM IN NEWFOUNDLAND, CANADA

The Marine Institute of Memorial University operates the SmartBay ocean observing system. SmartBay serves two principle needs. First it provides the marine stakeholders in the area with source of marine meteorological and oceanographic information in plain language and tailored to their particular application. Secondly, the system serves as a test platform to field test and allow for demonstration of new products; primarily new ocean sensors and renewable ocean energy systems.

The system collects data from a number of met/ocean buoys deployed in coastal waters around southeastern area of the island of Newfoundland. Data is gathered to a host server located at the Institute in St. John's, NL from where it is made available to the public as specific information products or as archived data depending on the end user application.

The SmartBay system serves as the host for the Marine Institute PUCK evaluation activities. The demonstration will be conducted on an operational platform consisting of a spar buoy equipped with an Axys Watchman 500 buoy controller located in Holyrood, Conception Bay approximately 50 km southwest of St. John's. The buoy is equipped with a standard marine weather station as well as selected oceanographic instruments and it connects to the Marine Institute host network via a WiFi radio link. Data from the demonstration is available through the SmartBay website www.smartbay.ca.

The demonstration illustrated the functionality of interoperable sensors operating through an Axys Watchman 500 buoy controller that utililzes the PUCK v1.3 protocol. For the purpose of this demonstration, two PUCK-enabled instruments were used - an RBR model XR-420 Conductivity/Temperature/Depth (CTD) probe and a Nortek Aquadopp Acoustic Doppler Current Profiler (ADCP). The XR-420 and the Aquadopp are cabled identically so they can both be easily plugged into the same serial port on the Axys Watchman buoy controller, operating on an Oceanographic buoy, connected to the SmartBay host.

The Watchman 500 buoy controller cycles through each and all serial ports and attempts to put an attached instrument into PUCK mode. If the attached instrument responds to PUCK protocol commands then the controller retrieves the PUCK datasheet which includes the instrument manufacturer code and loads the appropriate instrument driver from an onboard library. If a PUCK-enabled instrument is not detected on a serial port then the controller checks the next port until all ports have been checked will move on to the next serial port. After this point, the Watchman500 initializes all devices via their instrument drivers and goes into normal operation mode. The device driver will then proceed to collect data from the sensor and data will be available immediately after the sampling period is finished

The Watchman500 only does this on an initial boot up or a requested reset. Therefore, if new PUCK enabled sensors are installed in the field, the Watchman500 should be reset to have the new PUCK sensor initialized and used

CONCLUSIONS

PUCK Protocol can co-exist and it is compatible with other existing standards as IEEE1451 or SWE – SOS. The use of PUCK protocol with in an instrument facilitate the integration of the instrument within an observatory allowing storage of a self-descript instrument metadata in different payloads types as IEEE1451 XML TEDS or SensorML. The engineering effort required integrating a PUCK enable instrument into and observatory is very small. Within a working day a computer science engineer is able to understand and communicate with a PUCK enable instrument, storing and configuring its payload. Approximately one week is enough time to define the payload and generate the code to be ready to integrate the instrument into the observatory. An automatic instrument recognition protocol has been proposed in order to enable the host to automatically configure a new instrument using PUCK Protocol and different Payload types.

REFERENCES

[1] Lee, K., "IEEE 1451: A Standard in Support of Smart Transducer Networking", Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference, Baltimore, MD, May 1-4, 2000, Vol. 2, p.525-528

[2]George Percivall, Carl Reed "OGC® Sensor Web Enablement Standards", Sensors & Transducers Journal, Vol.71, Issue 9, September 2006, pp.698-706, ISSN 1726-5479

[3] Kent L. HEADLEY, Dan DAVIS, Duane EDGINGTON, Lance McBRIDE , Thomas C. O'REILLY, Michael RISI "Managing Sensor Network Configuration and Metadata in Ocean Observatories Using Instrument Pucks" Third International Workshop on Scientific Use of Submarine Cables and Related Technologies, 25-27 June 2003

[4] Marc Nogueras, Carola Artero, Joquín del Rio, Antoni Mànuel, David Sarrià, "Control and acquisition system design for an Expandable Seafloor Observatory", IEEE OCEANS09, 11-14 May, Bremen, Germany

[5]M. Nogueras, J. Santamaria, A. Mànuel, "Construction of the OBSEA cabled Submarine Observatory" Instrumentation Viewpoint, Num.6, pp33-34, autumn 2007.

[6] T.O'Reilly, K.Headley et al, "MBARI technology for self-configuring interoperable ocean observatories", Proceedings of the Marine Technology Society / Institute of Electrical and Electronics Engineers Oceans Conference, Boston, Massachusetts

[7] Rueda, C., Bermudez, L., Fredericks, J. The MMI Ontology Registry and Repository: A Portal for Marine Metadata Interoperability. MTS/IEEE Oceans'09. Biloxi, Mississippi. October, 2009.